IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 6 September 2022, accepted 29 September 2022, date of publication 3 October 2022, date of current version 11 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3211432

RESEARCH ARTICLE

Comparison Study of Iterative Algorithms for Center-of-Sets Type-Reduction of Takagi Sugeno Kang Type General Type-2 Fuzzy Logic Systems

YANG CHEN, (Member, IEEE)
College of Science, Liaoning University of Technology, Jinzhou 121001, China
e-mail: lxychenyang@lnut.edu.cn

This work was supported in part by the National Natural Science Foundation of China under Grant 61973146 and Grant 61773188, in part by the Doctoral Start-up Foundation of Liaoning Province under Grant 2021-BS-258, and in part by the Youth Fund of Education Department of Liaoning Province under Grant LJKQZ2021143.

ABSTRACT Studying the block of center-of-sets (COS) type-reduction (TR) for Takagi Sugeno Kang (TSK) inference-based general type-2 fuzzy logic systems (GT2 FLSs) is meaningful for applying the systems. Blocks of fuzzy reasoning, COS type-reduction and defuzzification for TSK type GT2 FLSs are first given. According to three kinds of iterative algorithms for computing the centroids of interval type-2 fuzzy sets (IT2 FSs), the paper extends these types of algorithms for studying the COS type-reduction of TSK type GT2 FLSs. Six computer simulation experiments show the computational costs of proposed three kinds of iterative algorithms by computing the outputs of GT2 FLSs, which affords the potential guidance value designers and users of T2 FLSs.

INDEX TERMS Iterative algorithms, Takagi Sugeno Kang inference, general type-2 fuzzy logic systems, computational cost, simulation.

I. INTRODUCTION

In contrast to the traditional type-1 fuzzy sets (T1 FSs), interval or general type-2 fuzzy sets (IT2 or GT2 FSs) can better model and cope with uncertainties by adjusting the additional parameters of footprint of uncertainty (FOU [1]). Therefore, T2 fuzzy logic systems (FLSs) based on T2 FSs [2], [3], [4], [5], [6], [7], [8] have become an emerging technology which has been successfully applied in many fields affected by high uncertainties, time-varying and nonlinearities. As shown in Fig. 1, the type-reduction (TR) plays a key role for T2 FLSs, which mainly maps the T2 FS to the T1 FS. Then the defuzzification transforms the T1 FS to the crisp output. While a T1 FLS does not have the TR block. Therefore, ~~the~~ computations in the former are much complicated than the latter, which makes the calculations in T2 FLSs more challenged.

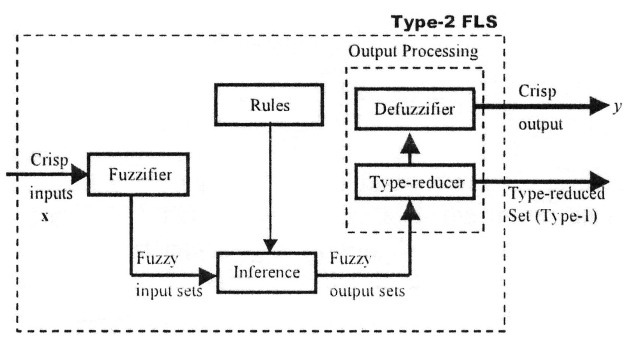

FIGURE 1. Modules of T2 FLSs.

As the secondary membership grades of IT2 FSs are all equal to one, the computational relatively simple IT2 FLSs [4], [9], [10], [11], [12] are currently most widely used T2 FLSs. However, since the alpha-planes or z-Slices representation [3], [8], [13], [14], [15] of GT2 FSs were proposed by a few different research groups, the computational

The associate editor coordinating the review of this manuscript and approving it for publication was Qi Zhou.

complexity of GT2 FLSs were greatly reduced. That is due to the fact that a GT2 FS can be decomposed into IT2 FSs based on alpha-planes or z-Slices. Then the design and applications of GT2 FLSs [7], [16], [17], [18], [19], [20], [21], [22], [23] based on GT2 FSs have gained rapidly development in the past decades.

In early days, the Karnik-Mendel (KM) algorithms [24], [25], [26] were put forward to compute the centroids of IT2 FSs. However, this type of computationally intensive algorithm is time consuming. Wu proposed the enhance KM (EKM) algorithms [27] to save the calculation time. Compared with the KM algorithms, the EKM algorithms can save about two iterations on average for computing the centroids. Another well-known type of non-KM iterative algorithm was also proposed, which was called as the enhanced iterative algorithms with stopping condition (EIASC [28]). Professor Jerry Mendel [3], [25] proposed that these three types of centroid type-reduction (TR) algorithms were good approaches for calculating the centroids. Despite so, the centroid TR methods focus on pure theoretical research. While the center-of-sets (COS) TR doesn't require that a complete description of a general type-2 fuzzy set be available as the centroid TR. Actually, studying the COS TR [29], [30], [31] is more helpful for applying T2 FLSs [32], [33], [34].

We arrange the rest of the paper as below. Section two briefly gives the background of Takagi Sugeno Kang (TSK) type GT2 FLSs. Three kinds of iterative algorithms for performing the COS TR of TSK type GT2 FLSs are provided in Section three. Section four shows six computer simulation examples to illustrate the performances of three types of iterative algorithms. Finally the conclusions are given in Section five.

II. TAKAGI SUGENO KANG TYPE GT2 FLSs

GT2 FLSs can usually be divided into Mamdani type and TSK type from the viewpoint of inference (fuzzy reasoning). Here we focus on the TSK type. Take into account a TSK type GT2 FLS with m inputs $x_1 \in X_1, \cdots, x_m \in X_m$ and a single output $y \in Y$. Without loss of generality, suppose that the nth fuzzy rule be as:

$$\tilde{R}^n : \text{ If } x_1 is \tilde{F}_1^n \text{ and } \cdots \text{ and } x_m \text{ } is \text{ } \tilde{F}_m^n,$$

$$\text{then } y_\alpha^n = c_0^n + \sum_{l=1}^{m} c_l^n x_l (n = 1, \cdots, N) \quad (1)$$

where $\tilde{F}_l^n (l = 1, \cdots, m)$ is the antecedent GT2 FS, in addition, $c_j^n (j = 0, \cdots, m)$ is the type-0 consequent parameter. Here we select the input measurement as the GT2 FS, and this type of TSK GT2 FLSs is the "$A2 - C0$", that is to say, the antecedent is the GT2 FS, and the consequent is the crisp (type-0) set.

First of all, we calculate the consequent output of each rule, i.e., $\{y_\alpha^n(x)\}_{n=1}^N$. Then it can be renumbered in an ascending order as: $\{\beta_\alpha^n(x)\}_{n=1}^N$. As for the non-singleton fuzzifier, the fired interval $A_\alpha^n(x)$ of each fuzzy rule under the alpha-level is as:

$$A_\alpha : \begin{cases} A_\alpha^n(x) & \equiv [\underline{a}_\alpha^n(x), \overline{a}_\alpha^n(x)] \\ \underline{a}_\alpha^n(x) & \equiv T_{l=1}^m \underline{a}_{l,\alpha}^n(x_l) = T_{l=1}^m \underline{u}_{\tilde{P}_{l,\alpha}^n}(\underline{x}_{l,\max}^n) \\ & = T_{l=1}^m [\underline{u}_{\tilde{X}_{l,\alpha}}(\underline{x}_{l.\max}^n) \wedge \underline{u}_{\tilde{F}_{l,\alpha}^n}(\underline{x}_{l.\max}^n)] \\ \overline{a}_\alpha^n(x) & \equiv T_{l=1}^m \overline{a}_{l,\alpha}^n(x_l) = T_{l=1}^m \overline{u}_{\tilde{P}_{l,\alpha}^n}(\underline{x}_{l,\max}^n) \\ & = T_{l=1}^m [\overline{u}_{\tilde{X}_{l,\alpha}}(\overline{x}_{l.\max}^n) \wedge \overline{u}_{\tilde{F}_{l,\alpha}^n}(\overline{x}_{l.\max}^n)] \end{cases} \quad (2)$$

here the $A_\alpha^n(x)$ is reorder as the order of $\{\beta_\alpha^n(x)\}_{n=1}^N$, and $\overline{x}_{l,\max}^n$, and $\underline{x}_{l,\max}^n$ are the x_i values that are corresponding to the $\sup_{x_l} \overline{u}_{\tilde{P}_{l,\alpha}^n}(x_l)$ and $\sup_{x_l} \underline{u}_{\tilde{P}_{l,\alpha}^n}(x_l)$, respectively.

As for the TSK type GT2 FLSs, whose output under the corresponding alpha-level can be computed as:

$$Y_{TSK,\alpha} = \alpha / [\frac{\sum_{n=1}^{N} A_\alpha^n(x) \beta_\alpha^n(x)}{\sum_{n=1}^{N} A_\alpha^n(x)}] = \alpha / [y_{l,\alpha}(x), y_{r,\alpha}(x)] \quad (3)$$

in which the two end points $y_{l,\alpha}(x)$ and $y_{r,\alpha}(x)$ can be calculated by several different kinds of TR algorithms as:

$$y_{l,\alpha}(x) = \min_{a_\alpha^n \in [\underline{a}_\alpha^n, \overline{a}_\alpha^n]} \frac{\sum_{n=1}^{N} a_\alpha^n(x) \underline{\beta}_\alpha^n(x)}{\sum_{n=1}^{N} a_\alpha^n(x)} \quad (4)$$

$$y_{r,\alpha}(x) = \max_{a_\alpha^n \in [\underline{a}_\alpha^n, \overline{a}_\alpha^n]} \frac{\sum_{n=1}^{N} a_\alpha^n(x) \overline{\beta}_\alpha^n(x)}{\sum_{n=1}^{N} a_\alpha^n(x)} \quad (5)$$

Here we make an assumption that the number of effective alpha-planes be k, i.e., the value of α can be equally divided into: $\alpha = \alpha_1, \alpha_2, \cdots, \alpha_k$. Then the final output of GT2 FLSs is as:

$$y_{TSK}(x) = \frac{\sum_{i=1}^{k} \frac{\alpha_i}{2} [y_{r,\alpha_i}(x) + y_{l,\alpha_i}(x)]}{\sum_{i=1}^{k} \alpha_i} \quad (6)$$

which was first proposed by Wagner and Hagras [8], and the equation (6) can be called as the average of end points defuzzification. This means we have to compute many $Y_{TSK,\alpha}$ values in terms of the corresponding α. The paper is aim at performing the COS TR for GT2 FLSs.

III. THREE KINDS OF ITERATIVE ALGORITHMS

Here the KM algorithms, EKM algorithms and EIASC are extended to complete the COS type-reduction of TSK inference structure-based GT2 FLSs.

A. KM ALGORITHMS

For the TSK type GT2 FLSs [3], [6], [19], let the COS type-reduced set under the related alpha-level be an interval, i.e.,

TABLE 1. Calculational steps for KM algorithms to perform the COS type-reduction of GT2 FLSs.

Step	KM algorithms to calculate $y_{l,\alpha}$
1	1. Initialize $\theta_{n,\alpha}$, set $\theta_{n,\alpha}=[\underline{a}_{\alpha}^{n}+\overline{a}_{\alpha}^{n})]/2,\ n=1,L\ ,N$, compute $c'=\dfrac{\sum_{n=1}^{N}\underline{\beta}_{\alpha}^{n}\theta_{n,\alpha}}{\sum_{n=1}^{N}\theta_{n,\alpha}}$
2	2. Find $L(1\le L\le N-1)$, which satisfies $\underline{\beta}_{\alpha}^{L}\le c'\le\underline{\beta}_{\alpha}^{L+1}$
3	3. When $n\le L$, set $\theta_{n,\alpha}=\overline{a}_{\alpha}^{n}$; when $n\ge L+1$, set $\theta_{n,\alpha}=\underline{a}_{\alpha}^{n}$, compute $y_{l,\alpha}(n)=\dfrac{\sum_{n=1}^{L}\underline{\beta}_{\alpha}^{n}\overline{a}_{\alpha}^{n}+\sum_{n=L+1}^{N}\underline{\beta}_{\alpha}^{n}\underline{a}_{\alpha}^{n}}{\sum_{n=1}^{L}\overline{a}_{\alpha}^{n}+\sum_{n=L+1}^{N}\underline{a}_{\alpha}^{n}}$
4	4. Check if $y_{l,\alpha}(n)=c'$, if so, stop and set $y_{l,\alpha}(n)=y_{l,\alpha},\ n=L$
5	5. Set $c'=y_{l,\alpha}(n)$ and return to step 2
Step	**KM algorithms to calculate $y_{r,\alpha}$**
1	1. The same as former in step 1, except for compute $c'=\dfrac{\sum_{n=1}^{N}\overline{\beta}_{\alpha}^{n}\theta_{n,\alpha}}{\sum_{n=1}^{N}\theta_{n,\alpha}}$
2	2. Find $R(1\le R\le N-1)$, which satisfies $\overline{\beta}_{\alpha}^{R}\le c'\le\overline{\beta}_{\alpha}^{R+1}$
3	3. When $n\le R$, set $\theta_{n,\alpha}=\underline{a}_{\alpha}^{n}$; when $n\ge R+1$, set $\theta_{n,\alpha}=\overline{a}_{\alpha}^{n}$, compute $y_{r,\alpha}(n)=\dfrac{\sum_{n=1}^{R}\overline{\beta}_{\alpha}^{n}\underline{a}_{\alpha}^{n}+\sum_{n=R+1}^{N}\overline{\beta}_{\alpha}^{n}\overline{a}_{\alpha}^{n}}{\sum_{n=1}^{R}\underline{a}_{\alpha}^{n}+\sum_{n=R+1}^{N}\overline{a}_{\alpha}^{n}}$
4	4. Check if $y_{r,\alpha}(n)=c'$, if so, stop and set $y_{r,\alpha}(n)=y_{r,\alpha},\ n=R$
5	5. Set $c'=y_{r,\alpha}(n)$ and return to step 2

$y_{TSK,\alpha}=[y_{l,\alpha},y_{r,\alpha}]$. Then the two end points $y_{l,\alpha}$ and $y_{r,\alpha}$ can be calculated in a non-closed form as:

$$y_{l,\alpha}(L)=\frac{\sum_{n=1}^{L}\underline{\beta}_{\alpha}^{n}\overline{a}_{\alpha}^{n}+\sum_{n=L+1}^{N}\underline{\beta}_{\alpha}^{n}\underline{a}_{\alpha}^{n}}{\sum_{n=1}^{L}\overline{a}_{\alpha}^{n}+\sum_{n=L+1}^{N}\underline{a}_{\alpha}^{n}}\approx y_{l,\alpha} \tag{7}$$

and

$$y_{r,\alpha}(R)=\frac{\sum_{n=1}^{R}\overline{\beta}_{\alpha}^{n}\underline{a}_{\alpha}^{n}+\sum_{n=R+1}^{N}\overline{\beta}_{\alpha}^{n}\overline{a}_{\alpha}^{n}}{\sum_{n=1}^{R}\underline{a}_{\alpha}^{n}+\sum_{n=R+1}^{N}\overline{a}_{\alpha}^{n}}\approx y_{r,\alpha} \tag{8}$$

where L and R are the left and right switching points.

Table 1 provides the specific calculational steps for KM algorithms to perform the COS type-reduction of GT2 FLSs. As for the equation (7), when $n=L+1$, $\theta_{n,\alpha}$ starts to change from the upper firing degree $\overline{a}_{\alpha}^{n}$ to the lower firing degree $\underline{a}_{\alpha}^{n}$; for the equation (8), when $n=R+1$, $\theta_{n,\alpha}$ starts to change from the lower firing degree $\underline{a}_{\alpha}^{n}$ to the upper firing degree $\overline{a}_{\alpha}^{n}$.

Then the COS defuzzified value at the related alpha-level can be computed as:

$$Y_{KM,\alpha}=(y_{l,\alpha}+y_{r,\alpha})/2. \tag{9}$$

Aggregating al ~~the~~ $Y_{KM,\alpha}$ to obtain the final type-reduced set Y_{KM}, i.e.,

$$Y_{KM}=\sup_{\forall\alpha\in[0,1]}\alpha/Y_{KM,\alpha}. \tag{10}$$

Finally the output of GT2 FLSs is as:

$$y_{KM}=\frac{\sum_{i=1}^{k}\alpha_i\{[y_{l,\alpha_i}(x)+y_{r,\alpha_i}(x)]/2\}}{\sum_{i=1}^{k}\alpha_i} \tag{11}$$

B. EKM ALGORITHMS

In fact, the EKM algorithms [25], [27], [32], [33] generate from the KM algorithms. Despite so, the EKM algorithms improve the KM algorithms in three points as:

1) a better initialization approach is provided for saving the iterations; 2) according to the stopping condition, the unnecessary iteration is cancelled; 3) in terms of a smart calculation technique, the computational cost is reduced.

Table 2 provides the specific calculational steps for EKM algorithms to perform the COS type-reduction of GT2 FLSs.

Then the COS defuzzified value for EKM algorithms at the related alpha-level can be computed as:

$$Y_{EKM,\alpha}=(y_{l,\alpha}+y_{r,\alpha})/2. \tag{12}$$

Aggregating all the $Y_{EKM,\alpha}$ to get the final type-reduced set Y_{EKM}, i.e.,

$$Y_{EKM}=\sup_{\forall\alpha\in[0,1]}\alpha/Y_{EKM,\alpha}. \tag{13}$$

Finally the output of GT2 FLSs for EKM algorithms is as:

$$y_{EKM}=\frac{\sum_{i=1}^{k}\alpha_i\{[y_{l,\alpha_i}(x)+y_{r,\alpha_i}(x)]/2\}}{\sum_{i=1}^{k}\alpha_i}. \tag{14}$$

C. EIASC

The non-KM type of EIASC is a type of iterative algorithm which originates from the monotoncity and shapes of IT2 FSs is comparatively easy to understand. For the equation (7), here we let $y_{l,\alpha}(L)$ be first monotone decreasing as L increases, then it should be monotone increasing as L increases. And for the equation (8), here we let $y_{r,\alpha}(R)$ be first monotone increasing as R increases, then it should be monotone decreasing as R increases. Table 3 provides the specific calculational steps for EIASC to perform the COS type-reduction of GT2 FLSs.

TABLE 2. Calculational steps for EKM algorithms to perform the COS type-reduction of GT2 FLSs.

Step	EKM algorithms to compute $y_{l,\alpha}$
1	1. Set $s=[N/2.4]$ (the closest integer to $N/2.4$) and calculate $a=\sum_{n=1}^{s}\underline{\beta}^{n}_{\alpha}\overline{a}^{n}_{\alpha}+\sum_{n=s+1}^{N}\underline{\beta}^{n}_{\alpha}\underline{a}^{n}_{\alpha}$, $b=\sum_{n=1}^{k}\overline{a}^{n}_{\alpha}+\sum_{n=s+1}^{N}\underline{a}^{n}_{\alpha}$, $c'=a/b$
2	2. Find $s'\in[1,N-1]$, which satisfies $\underline{\beta}^{s'}_{\alpha}\le c'\le\underline{\beta}^{s'+1}_{\alpha}$
3	3. Check if $s'=s$, if so, stop and set $c'=y_{l,\alpha}$, $s=L$; otherwise, go to step 4
4	4. Compute $l=sign(s'-s)$, $a'=a+l\sum_{n=\min(s,s')+1}^{\max(s,s')}\underline{\beta}^{n}_{\alpha}(\overline{a}^{n}_{\alpha}-\underline{a}^{n}_{\alpha})$, $b'=b+l\sum_{n=\min(s,s')+1}^{\max(s,s')}(\overline{a}^{n}_{\alpha}-\underline{a}^{n}_{\alpha})$, $c''(s')=a'/b'$
5	5. Set $c'=c''(s), a=a'$, and $b=b'$ and return to step 2
Step	EKM algorithms to compute $y_{r,\alpha}$
1	1. Set $s=[N/1.7]$ (the closest integer to $N/1.7$) and calculate $a=\sum_{n=1}^{s}\overline{\beta}^{n}_{\alpha}\underline{a}^{n}_{\alpha}+\sum_{n=s+1}^{N}\overline{\beta}^{n}_{\alpha}\overline{a}^{n}_{\alpha}$, $b=\sum_{n=1}^{s}\underline{a}^{n}_{\alpha}+\sum_{n=s+1}^{N}\overline{a}^{n}_{\alpha}$, $c'=a/b$
2	2. Find $s'\in[1,N-1]$, which satisfies $\overline{\beta}^{s'}_{\alpha}\le c'\le\overline{\beta}^{s'+1}_{\alpha}$
3	3. The same as the former in step 3, except for setting $c'=y_{r,\alpha}$ and $s=R$
4	4. Compute $l=sign(s'-s)$, $a'=a-l\sum_{n=\min(s,s')+1}^{\max(s,s')}\overline{\beta}^{n}_{\alpha}(\overline{a}^{n}_{\alpha}-\underline{a}^{n}_{\alpha})$, $b'=b-l\sum_{n=\min(s,s')+1}^{\max(s,s')}(\overline{a}^{n}_{\alpha}-\underline{a}^{n}_{\alpha})$, and $c''(s')=a'/b'$
5	5. The same as the former in step 5

Then the COS defuzzified value for EIASC at the related alpha-level can be computed as:

$$Y_{EIASC,\alpha}=(y_{l,\alpha}+y_{r,\alpha})/2. \tag{15}$$

Aggregating all $Y_{EIASC,\alpha}$ to obtain the final type-reduced set Y_{EIASC}, i.e.,

$$Y_{EIASC}=\sup_{\forall\alpha\in[0,1]}\alpha/Y_{EIASC,\alpha}. \tag{16}$$

Finally the output of GT2 FLSs can be as:

$$y_{EIASC}=\frac{\sum_{i=1}^{k}\alpha_i\{[y_{l,\alpha_i}(x)+y_{r,\alpha_i}(x)]/2\}}{\sum_{i=1}^{k}\alpha_i}. \tag{17}$$

IV. SIMULATION EXPERIMENTS

Here six simulation instances are used to show how to adopt the three kinds of iterative algorithms to perform the COS type-reduction and defuzzification of TSK inference based GT2 FLSs. In the simulations, the value of α is equally divided into Δ effective values as: $\alpha=0,1/\Delta,\cdots,1$. In addition, let the Δ be changed from 1 to the maximum number 100 with the stepsize of 1. Here the COS type-reduced set as $\Delta=100$ and the COS defuzzified values as $\Delta=1:1:100$ computed by three types of iterative algorithms are investigated.

TABLE 3. Calculational steps for EIASC to perform the COS type-reduction of GT2 FLSs.

Step	EIASC to compute $y_{l,\alpha}$
1	Initialization: $a=\sum_{n=1}^{N}\underline{\beta}^{n}_{\alpha}\underline{a}^{n}_{\alpha}$, $b=\sum_{n=1}^{N}\underline{a}^{n}_{\alpha}$, $L=0$
2	Compute: $L=L+1$; $a=a+\underline{\beta}^{L}_{\alpha}(\overline{a}^{L}_{\alpha}-\underline{a}^{L}_{\alpha})$; $b=b+(\overline{a}^{L}-\underline{a}^{L})$; $y_{l,\alpha}=a/b$
3	If $y_{l,\alpha}\le\beta^{L+1}_{\alpha}$, stop; otherwise, return to step 2
	EIASC to compute $y_{r,\alpha}$
1	Initialization: $a=\sum_{n=1}^{N}\overline{\beta}^{n}_{\alpha}\overline{a}^{n}_{\alpha}$, $b=\sum_{n=1}^{N}\overline{a}^{n}_{\alpha}$, $R=N$
2	Compute: $a=a+\overline{\beta}^{R}_{\alpha}(\overline{a}^{R}_{\alpha}-\underline{a}^{R}_{\alpha})$; $b=b+(\overline{a}^{R}_{\alpha}-\underline{a}^{R}_{\alpha})$; $y_{r,\alpha}=a/b$; $R=R-1$
3	If $y_{r,\alpha}\ge\beta^{R}_{\alpha}$, stop; otherwise, return to step 2

In the simulations example 1 and example 2, let each fuzzy rule be described by 4 antecedents and 1 consequent. Furthermore, suppose that 2 GT2 FSs are used for characterizing each antecedent. So that, there are totally 2^4, i.e., sixteen fuzzy rules for the TSK inference based GT2 FLSs. For the *nth* fuzzy rule, whose form can be as:

$$\text{If } x_1 \text{ is } \tilde{F}_1^n \text{ and } \ldots \text{ and } x_4 \text{ is } \tilde{F}_4^n, \quad \text{then } Y^n=C_0^n+\sum_{l=1}^{4}C_l^n x_l \tag{18}$$

where $\tilde{F}_l^n(l=1,\cdots,4;n=1,\cdots,16)$ denotes the antecedent GT2 FS, $C_i^n(i=0,\cdots,4;n=1,\cdots,16)$ represents the consequent, and $C_i^n=[c_i^n-s_i^n,c_i^n+s_i^n]$.

Example 1: As for the TSK inference based GT2 FLSs, the primary MF of antecedent GT2 FS is chosen as the Gaussian type MF, i.e., see the Figure 2,

$$\mu_l^n(x_l)=\exp[-\frac{1}{2}(\frac{x_l-m_l^n}{\sigma_l^n})^2] \quad (l=1,\cdots,4;n=1,\cdots,16) \tag{19}$$

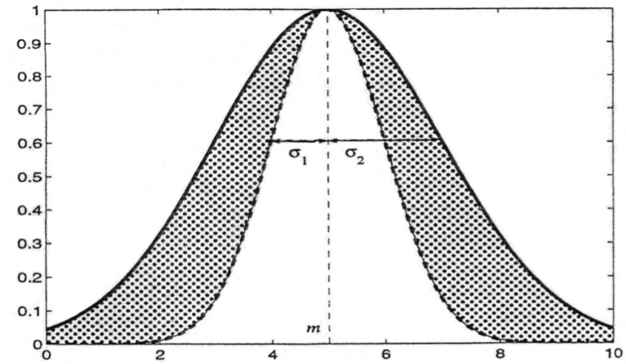

FIGURE 2. Shape of FOU of gaussian GT2 fuzzy set with uncertain standard deviation.

in which $\sigma_l^n \in [\sigma_{l1}^n, \sigma_{l2}^n]$, and the secondary membership function is selected as the triangular type MF, that is to say,

$$Apex = u_1(x) + [u_2(x) - u_1(x)]/5 \tag{20}$$

in which $u_2(x)$ and $u_1(x)$ represent the upper bound and lower bound for FOU, respectively.

Then the parameters for antecedents are selected as:

$$\sigma_{l1}^n = 2.2 + rand(4, 16) \tag{21}$$
$$\sigma_{l2}^n = \sigma_{l1}^n + rand(4, 16) \tag{22}$$
$$m_l^n = 1.8 + 2 * rand(4, 16). \tag{23}$$

In addition, the parameters for C_i^n are selected as:

$$c_i^n = rand(1, 4), \quad s_i^n = rand(1, 4)(i = 0, 1, \cdots, 4) \tag{24}$$

The input measurement set is chosen as:

$$x = 7 * rand(16, 4). \tag{25}$$

Example 2: For the TSK inference based GT2 FLSs, the primary MF of antecedent GT2 FS is chosen as the Gaussian type MF with uncertain mean, i.e., see the Figure 3,

$$\mu_l^n(x_l) = \exp\{-\frac{1}{2}(\frac{x_l - m_l^n}{\sigma_l^s})^2\} \quad (l = 1, \cdots, 4; n = 1, \cdots, 16) \tag{26}$$

in which $m_l^n \in [m_{l1}^n, m_{l2}^n]$ and the secondary membership function is selected as the trapezoidal type MF, that is to say,

$$L(x) = u_1(x) \tag{27}$$
$$R(x) = u_2(x) - 3[u_2(x) - u_1(x)]/5. \tag{28}$$

Then the parameters for antecedents are selected as:

$$m_{l1}^n = 1.7 + rand(4, 16) \tag{29}$$
$$m_{l2}^n = m_{l1}^n + 2 * rand(4, 16) \tag{30}$$
$$\sigma_l^n = 2.3 + rand(4, 16) \tag{31}$$

In addition, the parameters for C_i^n and input measurement are selected the same form as in equation (24) and equation (25), respectively.

As for the example 3 and example 4, the forms of all parameters of TSK inference based GT2 FLSs are selected the same as in 1 examples 1 and example 2, respectively. However, the number of antecedents in each fuzzy rule is chosen as 5. Therefore, the number of fuzzy rules will be 2^5, that is to say, 32 in these two examples. While for the example 5 and example 6, the number of antecedents in each fuzzy rule is chosen as 6. Therefore, the number of fuzzy rules will be 2^6, that is to say, 64 in the last two examples. Next we perform both the quantitative and qualitative studies. For these six examples, as $\Delta = 100$, the COS type-reduced sets calculated by three types of iterative algorithms are provided in Figure 4.

As Δ be changed from 1 to the maximum number 100 with the stepsize of 1, the COS defuzzified outputs calculated by three types of iterative algorithms are provided in Figure 5.

Next the calculational times of three kinds of iterative algorithms are investigated. The hardware and software platforms

TABLE 4. Calculational time for obtaining the COS type-reduced sets.

Num	KM	EKM	EIASC	TRR_{EKM} (%)	TRR_{EIASC} (%)
Exampl	0.020211	0.012806	0.009510	36.64	52.95
Example 2	0.039075	0.012139	0.009608	68.93	75.41
Example 3	0.112328	0.045580	0.031706	59.42	71.77
Example 4	0.121255	0.042345	0.035999	65.08	70.31
Example 5	0.343352	0.077276	0.059383	*77.49*	*82.70*
Example 6	0.213085	0.055740	0.049467	73.84	76.79
Average	0.141551	0.040981	0.032612	71.05	76.96

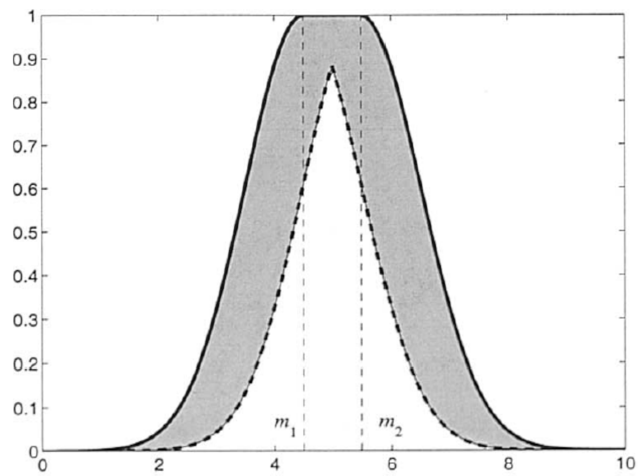

FIGURE 3. Shape of FOU of gaussian GT2 fuzzy set with uncertain mean.

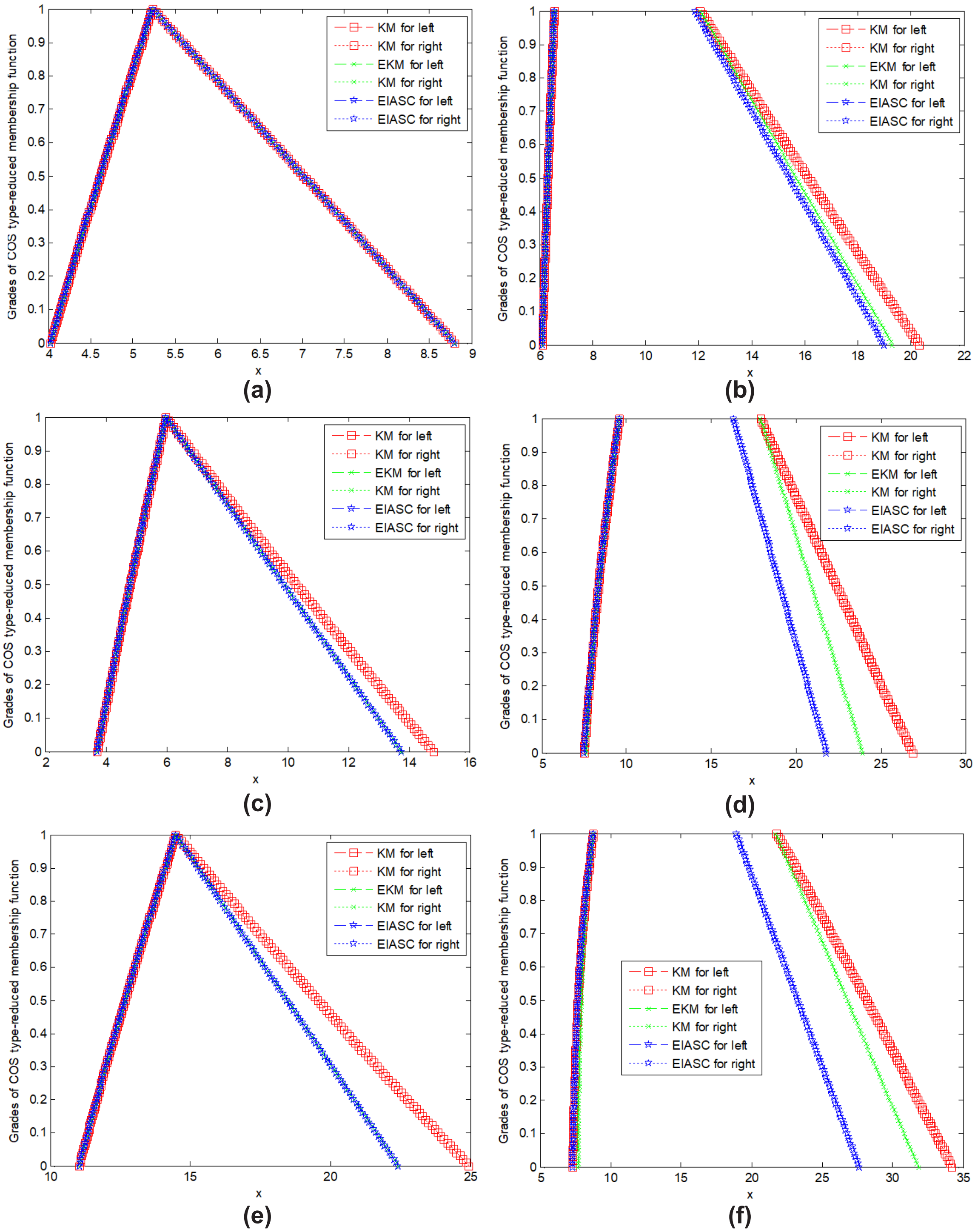

FIGURE 4. The obtained COS type-reduced sets, (a) example 1, (b) example 2, (c) example 3, (d) example 4, (e) example 5, and (f) example 6.

are chosen as a dell desktop with dual core CPU and the Matlab 2013a, respectively. In order to measure the efficiencies of these three types of iterative algorithms, the specific calculational times for obtaining the COS type-reduced sets and defuzzified values are provided in the following Table 4 and Table 5, respectively. Here we select the time unit as the second (s). Furthermore, the last two columns denote the time reducing rate (TRR) for the EKM algorithms and EIASC compared with the KM algorithms, respectively, and the last line in Table 1 and Table 2 represents the mean of 6 examples. The TRR is defined as:

$$TRR_{EKM,EIASC} = \frac{t_{KM} - t_{EKM,EIASC}}{t_{KM}} \times 100\% \qquad (32)$$

where t is the computational time of iterative algorithm.

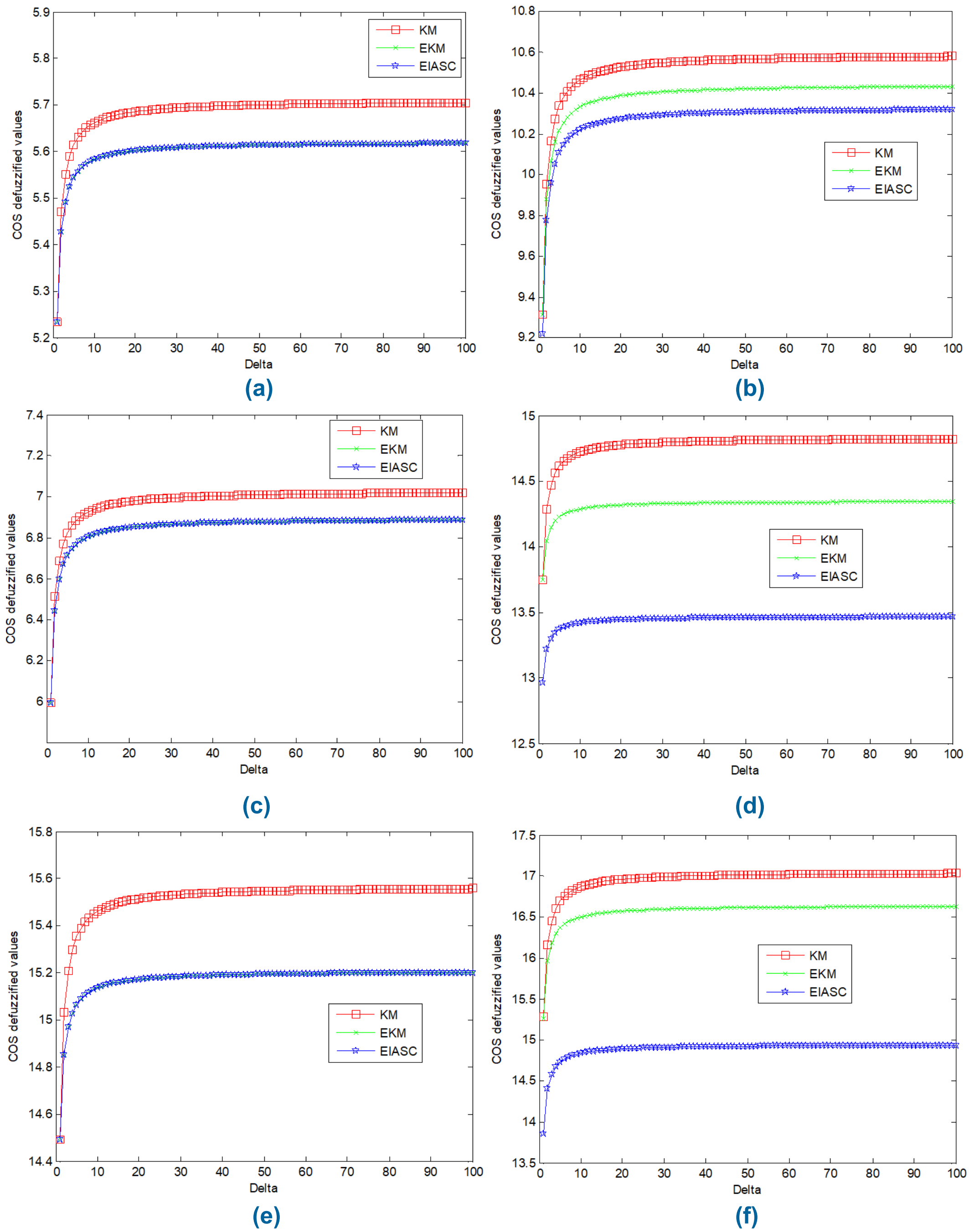

FIGURE 5. **The obtained COS defuzzified outputs, (a) example 1, (b) example 2, (c) example 3, (d) example 4, (e) example 5, and (f) example 6.**

Observing from the Figures 4-5 and Tables 4-5, the quantitative and qualitative conclusions for the provided six examples can be made:

1) For the left parts of COS type-reduced sets (see the Figure 4), from the first example to the last one, the simulation results of three types of iterative algorithms are almost completely the same; for the right parts of COS type-reduced sets, the results of KM algorithms, EKM algorithms and EIASC are slightly different, while the results two formers are almost completely the same in Examples 1, 3, and 5.

2) For the COS defuzzified outputs (see the Figure 5), the simulation results of KM algorithms, EKM algorithms and EIASC are slightly different, while the results two formers are almost completely the same in Examples 1, 3, and 5.

3) Compared with the KM algorithms, in these six examples, the EKM algorithms and EIASC obtain the maximum TRR values as 77.49%, and 82.70% for getting the COS type-reduced sets; and they obtain the maximum TRR values as 72.78%, and 73.78% for getting the COS defuzzified values.

4) In contrast to the KM algorithms, the EKM algorithms and EIASC obtain the average value of TRRs as 71.05%,

TABLE 5. Calculational time for obtaining the defuzzified values.

Num	KM	EKM	EIASC	TRR$_{EKM}$ (%)	TRR$_{EIASC}$ (%)
Exampl	1.850922	0.621958	0.536681	66.40	71.00
Example 2	1.896580	0.578742	0.536939	69.49	71.69
Example 3	4.530576	1.318976	1.275853	70.89	71.84
Example 4	4.796805	1.334434	1.257821	72.18	*73.78*
Example 5	10.506645	2.941391	2.849238	72.00	72.88
Example 6	10.694402	2.911517	2.840023	*72.78*	73.44
Average	5.712655	1.617836	1.549426	71.68	72.88

and 76.96% for getting the COS type-reduced sets; and they obtain the average value of TRRs as 71.68%, and 72.88% for getting the COS defuzzified values.

Here we put forward the three kinds of iterative algorithms for performing the COS TR and defuzzification of TSK inference based GT2 FLSs by means of six computer simulation examples. It can be shown that the calculational times of three kinds of iterative algorithms are gradually decreased, that is to say, the computational efficiencies are improved. Therefore, we may improve the iterative types of algorithms to investigate the COS TR of GT2 FLSs.

V. CONCLUSION AND EXPECTATIONS

Three kinds of iterative algorithms for performing the COS TR of TSK inference based GT2 FLSs are proposed in this paper. Furthermore, we provide the blocks of fuzzy reasoning, COS TR and defuzzification for GT2 FLSs. Then six computer simulation examples to used to illustrate the computational values and times of COS type-reduced sets and defuzzified for three types of iterative algorithms. Simulation results shows that the efficiencies of three types of iterative algorithms are gradually increased, which may be meaningful for designing GT2 FLSs in fields influenced by the uncertainties.

In the next, the COS TR of both IT2 FLSs and GT2 FLSs based on both noniterative algorithms and iterative algorithms [30], [35], [36], [37], [38], [39], [40], [41] will be further studies. In addition, we will investigate the applications of IT2 and GT2 fuzzy neural networks optimized with the combination of different types of intelligent algorithms [42], [43], [44], [45], [46] in forecasting, fuzzy control, and fuzzy identification and so on.

ACKNOWLEDGMENT

The author is grateful to the well-known Academician Jerry Mendel, who has provided many invaluable advices.

REFERENCES

[1] H. Mo, F.-Y. Wang, M. Zhou, R. Li, and Z. Xiao, "Footprint of uncertainty for type-2 fuzzy sets," *Inf. Sci.*, vol. 272, pp. 96–110, Jul. 2014.
[2] D. Wu and J. M. Mendel, "Recommendations on designing practical interval type-2 fuzzy systems," *Eng. Appl. Artif. Intell.*, vol. 85, pp. 182–193, Oct. 2019.
[3] J. M. Mendel, "General type-2 fuzzy logic systems made simple: A tutorial," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 5, pp. 1162–1182, Oct. 2014.
[4] Y. Chen, D. Wang, and S. Tong, "Forecasting studies by designing Mamdani interval type-2 fuzzy logic systems: With the combination of BP algorithms and KM algorithms," *Neurocomputing*, vol. 174, pp. 1133–1146, Jan. 2016.
[5] A. T. Azar, "Overview of type-2 fuzzy logic systems," *Int. J. Fuzzy Syst. Appl.*, vol. 2, no. 4, pp. 1–28, Oct. 2012.
[6] A. Khosravi and S. Nahavandi, "Load forecasting using interval type-2 fuzzy logic systems: Optimal type reduction," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1055–1063, May 2014.
[7] E. Ontiveros, P. Melin, and O. Castillo, "Comparative study of interval type-2 and general type-2 fuzzy systems in medical diagnosis," *Inf. Sci.*, vol. 525, pp. 37–53, Jul. 2020.
[8] C. Wagner and H. Hagras, "Toward general type-2 fuzzy logic systems based on zSlices," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 4, pp. 637–660, Aug. 2010.
[9] J. Mendel and X. Liu, "Simplified interval type-2 fuzzy logic systems," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 6, pp. 1056–1069, Dec. 2013.
[10] C. H. Hsu and C. F. Juang, "Evolutionary robot wall-following control using type-2 fuzzy controller with species-DE-activated continuous ACO," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 1, pp. 100–112, Feb. 2013.
[11] E. Ontiveros-Robles, P. Melin, and O. Castillo, "Comparative analysis of noise robustness of type 2 fuzzy logic controllers," *Kybernetika*, vol. 54, no. 1, pp. 175–201, Mar. 2018.
[12] D. Wang and Y. Chen, "Study on permanent magnetic drive forecasting by designing Takagi Sugeno Kang type interval type-2 fuzzy logic systems," *Trans. Inst. Meas. Control*, vol. 40, no. 6, pp. 2011–2023, Apr. 2018.
[13] F. Liu, "An efficient centroid type-reduction strategy for general type-2 fuzzy logic system," *Inf. Sci.*, vol. 178, no. 9, pp. 2224–2236, May 2008.
[14] J. M. Mendel, F. L. Liu, and D. Y. Zhai, "α-plane representation for type-2 fuzzy sets: Theory and applications," *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 5, pp. 1189–1207, Oct. 2009.
[15] E. Ontiveros, P. Melin, and O. Castillo, "High order α-planes integration: A new approach to computational cost reduction of general type-2 fuzzy systems," *Eng. Appl. Artif. Intell.*, vol. 74, pp. 186–197, Sep. 2018.
[16] P. Melin, C. I. Gonzalez, J. R. Castro, O. Mendoza, and O. Castillo, "Edge-detection method for image processing based on generalized type-2 fuzzy logic," *IEEE Trans. Fuzzy Syst.*, vol. 22, no. 6, pp. 1515–1525, Dec. 2014.
[17] C. I. Gonzalez, P. Melin, J. R. Castro, O. Mendoza, and O. Castillo, "An improved Sobel edge detection method based on generalized type-2 fuzzy logic," *Soft Comput.*, vol. 20, no. 2, pp. 773–784, Feb. 2016.
[18] Y. Chen and D. Wang, "Forecasting by designing Mamdani general type-2 fuzzy logic systems optimized with quantum particle swarm optimization algorithms," *Trans. Inst. Meas. Control*, vol. 41, no. 10, pp. 2886–2896, Jun. 2019.
[19] Y. Chen, D. Wang, and W. Ning, "Forecasting by TSK general type-2 fuzzy logic systems optimized with genetic algorithms," *Optim. Control Appl. Methods*, vol. 39, no. 1, pp. 393–409, Jan. 2018.
[20] Y. Chen and D. Wang, "Forecasting by general type-2 fuzzy logic systems optimized with QPSO algorithms," *Int. J. Control Automat. Syst.*, vol. 15, no. 6, pp. 2950–2958, Dec. 2017.
[21] D. Bernardo, H. Hagras, and E. Tsang, "A genetic type-2 fuzzy logic based system for the generation of summarised linguistic predictive models for financial applications," *Soft Comput.*, vol. 17, no. 12, pp. 2185–2201, Aug. 2013.
[22] M. A. Sanchez, O. Castillo, and J. R. Castro, "Generalized type-2 fuzzy systems for controlling a mobile robot and a performance comparison with interval type-2 and type-1 fuzzy systems," *Expert Syst. Appl.*, vol. 42, no. 14, pp. 5904–5914, Aug. 2015.
[23] O. Castillo, L. Amador-Angulo, J. R. Castro, and M. Garcia-Valdez, "A comparative study of type-1 fuzzy logic systems, interval type-2 fuzzy logic systems and generalized type-2 fuzzy logic systems in control problems," *Inf. Sci.*, vol. 354, pp. 257–274, Aug. 2016.
[24] N. N. Karnik and J. M. Mendel, "Centroid of a type-2 fuzzy set," *Inf. Sci.*, vol. 132, no. 1, pp. 195–220, Feb. 2001.
[25] J. M. Mendel, "On KM algorithms for solving type-2 fuzzy set problems," *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 3, pp. 426–446, Jun. 2013.

[26] J. M. Mendel and F. Liu, ''Super-exponential convergence of the Karnik–Mendel algorithms for computing the centroid of an interval type-2 fuzzy set,'' *IEEE Trans. Fuzzy Syst.*, vol. 15, no. 2, pp. 309–320, Apr. 2007.
[27] D. Wu and J. M. Mendel, ''Enhanced Karnik–Mendel algorithms,'' *IEEE Trans. Fuzzy Syst.*, vol. 17, no. 4, pp. 923–934, Aug. 2009.
[28] D. Wu and M. Nie, ''Comparison and practical implementation of type-reduction algorithms for type-2 fuzzy sets and systems,'' in *Proc. IEEE Int. Conf. Fuzzy Syst. (FUZZ-IEEE)*, Taiwan, Jun. 2011, pp. 2131–2138.
[29] Y. Chen and J. X. Yang, ''Comparison studies of iterative algorithms for center-of-sets type-reduction of interval type-2 fuzzy logic systems,'' *Int. J. Innov. Comput., Inf. Control*, vol. 18, no. 1, pp. 29–39, Jan. 2022.
[30] Y. Chen and J. Yang, ''Study on center-of-sets type-reduction of interval type-2 fuzzy logic systems with noniterative algorithms,'' *J. Intell. Fuzzy Syst.*, vol. 40, no. 6, pp. 11099–11106, Jun. 2021.
[31] D. Wu, ''Approaches for reducing the computational cost of interval type-2 fuzzy logic systems: Overview and comparisons,'' *IEEE Trans. Fuzzy Syst.*, vol. 21, no. 1, pp. 80–99, Feb. 2013.
[32] Y. Chen and D. Wang, ''Study on centroid type-reduction of general type-2 fuzzy logic systems with weighted enhanced Karnik–Mendel algorithms,'' *Soft Comput.*, vol. 22, no. 4, pp. 1361–1380, Feb. 2018.
[33] X. Liu, J. Mendel, and D. Wu, ''Study on enhanced Karnik–Mendel algorithms: Initialization explanations and computation improvements,'' *Inf. Sci.*, vol. 184, no. 1, pp. 75–91, Feb. 2012.
[34] Y. Chen, ''Study on weighted-based discrete noniterative algorithms for computing the centroids of general type-2 fuzzy sets,'' *Int. J. Fuzzy Syst.*, vol. 24, no. 1, pp. 587–606, Feb. 2022.
[35] Y. Chen, C. Li, and J. Yang, ''Design of discrete noniterative algorithms for center-of-sets type reduction of general type-2 fuzzy logic systems,'' *Int. J. Fuzzy Syst.*, vol. 24, no. 4, pp. 2024–2035, Jun. 2022.
[36] L. Wu, F. Qian, M. Wang, and T. Shang, ''An improved type-reduction algorithm for general type-2 fuzzy sets,'' *Inf. Sci.*, vol. 593, pp. 99–120, May 2022.
[37] L. Wu, F. Qian, M. Wang, and T. Shang, ''Improvement of enhanced opposite direction searching algorithm,'' *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 4, pp. 978–989, Apr. 2022, doi: 10.1109/TFUZZ.2021.3051333.
[38] S. Jianzhong, L. Shaohua, Y. Yong, and L. Rong, ''An improved general type-2 fuzzy sets type reduction and its application in general type-2 fuzzy controller design,'' *Soft Comput.*, vol. 23, no. 24, pp. 13513–13530, Mar. 2019.
[39] F. Gaxiola, P. Melin, F. Valdez, J. R. Castro, and O. Castillo, ''Optimization of type-2 fuzzy weights in backpropagation learning for neural networks using GAs and PSO,'' *Appl. Soft Comput.*, vol. 38, pp. 860–871, Jan. 2016.
[40] T. Zhao and J. Xiao, ''State feedback control of interval type-2 Takagi–Sugeno fuzzy systems via interval type-2 regional switching fuzzy controllers,'' *Int. J. Syst. Sci.*, vol. 46, no. 15, pp. 2756–2769, Nov. 2015.
[41] Q.-F. Fan, T. Wang, Y. Chen, and Z.-F. Zhang, ''Design and application of interval type-2 TSK fuzzy logic system based on QPSO algorithm,'' *Int. J. Fuzzy Syst.*, vol. 20, no. 3, pp. 835–846, Mar. 2018.
[42] A. Banakar and M. F. Azeem, ''Parameter identification of TSK neuro-fuzzy models,'' *Fuzzy Sets Syst.*, vol. 179, no. 1, pp. 62–82, Sep. 2011.
[43] C. K. Ahn, ''Some new results on stability of Takagi–Sugeno fuzzy Hopfield neural networks,'' *Fuzzy Sets Syst.*, vol. 179, no. 1, pp. 100–111, Sep. 2011.
[44] H. Liang, Z. Du, T. Huang, and Y. Pan, ''Neuroadaptive performance guaranteed control for multiagent systems with power integrators and unknown measurement sensitivity,'' *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 29, 2022, doi: 10.1109/TNNLS.2022.3160532.
[45] Y. Pan, Y. Wu, and H.-K. Lam, ''Security-based fuzzy control for nonlinear networked control systems with DoS attacks via a resilient event-triggered scheme,'' *IEEE Trans. Fuzzy Syst.*, early access, Feb. 7, 2022, doi: 10.1109/TFUZZ.2022.3148875.
[46] Y. Pan, Q. Li, H. Liang, and H.-K. Lam, ''A novel mixed control approach for fuzzy systems via membership functions online learning policy,'' *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 9, pp. 3812–3822, Sep. 2022.

YANG CHEN (Member, IEEE) received the B.S. and M.S. (in applied mathematics) degrees from the Liaoning University of Technology, China, in 2004 and 2007, respectively, and the Ph.D. degree in control theory and control engineering from Northeastern University, Shenyang, China, in 2020. Since 2019, he has been an Associate Professor with the Liaoning University of Technology. His research interests include fuzzy reasoning and fuzzy control, type-2 fuzzy logic systems, type-2 fuzzy neural networks, intelligent computing, and permanent magnetic drive.